Ordinary Membrane Machines versus Other Mathematical Models of Systems Realizing Massively Parallel Computations

Adam Obtułowicz

Institute of Mathematics of the Polish Academy of Sciences
E-mail: adamo@impan.gov.pl

Summary. A comparison of ordinary membrane machines, understood as certain recursive families of deterministic P systems, with some other mathematical models of systems realizing massively parallel computations is discussed. These mathematical models are those which respect recursiveness of computational tasks of systems, i.e., the functions to be computed are recursive functions and the decision problems correspond to recursive sets. The comparison together with open problems is summarized in the enclosed tables, where open problems are indicated by question mark "?".

1 Introduction

We present and discuss a comparison of ordinary membrane machines, understood as certain recursive families of deterministic P systems (for P systems see [23]), with some other mathematical models of systems realizing massively parallel computations. These mathematical models are those which respect recursiveness of computational tasks of systems, i.e., the functions to be computed are recursive functions and the decision problems correspond to recursive sets.

The comparison is discussed with regard to those (comparative) features of the mathematical models which one can treat as advantageous features from the logical or complexity theoretical point of view, or by means that they provide natural extensions of the (classes of) models for application of other approaches to computing than the discrete time approach or deterministic approach. The mathematical models are chosen in such a way that for every comparative feature there is provided at least one representative or typical example of a model of this feature.

2 Compared Models and Their Features

We discuss the following mathematical models of systems realizing massively parallel computations.

1: *ordinary membrane machines* defined to be recursive families $\Pi = (\Pi^i : i \in \mathrm{INP})$ of deterministic P systems Π^i for recursive sets INP of input data, where P systems Π^i are constructs understood as in [23] and their determinism is understood as in e.g. [16]. For more explanations see Remark 1 below;
2: Parallel Random Access Machines (PRAMs), cf. [9], [14], [20];
3: neural net models due to H. T. Siegelmann and E. D. Sontag, cf. [27];
4: R. Gandy's machines, cf. [10] and [26], the parallelism of their computations was pointed out in [7];
5: parallel Abstract State Machines and intra-step interacting Abstract State Machines due to Y. Gurevich, cf. [2];
6: Connection Machines due to W. D. Hillis, cf. [12].

The above models are such that they respect recursiveness of computational tasks understood as in Introduction.

The ordinary membrane machines require more explanations which are given in the following remark.

Remark 1 A representative example of an ordinary membrane machine is discussed in [18], where input data in INP are propositional formulas Φ in conjunctive normal form and the deterministic P systems Π^Φ—the elements of the family are used to solve SAT problem in a polynomial time, like in [16]. More precisely, the P system Π^Φ associated to a formula Φ generates that unique evolution process of membrane systems which provides a decision in a polynomial time (with respect to the number of clauses and the number of variables occurring in Φ) whether Φ holds for some valuation of variables occurring in Φ. The above recursive family of P systems was introduced in [18] to describe in a program-like uniform way the P systems solving SAT problem in [16].

We use "membrane machines" to name the families in 1 because evolving membrane systems are basic mechanisms of computations realized by P systems, see [23]. The adjective "ordinary" is applied to distinguish the families in 1 from other possible families of P systems, e.g. families of stochastic P systems or quantum P systems.

Remark 2 An evolving membrane system or simply a *membrane system* $\mathcal{S}$ is understood in the paper to be given by its *underlying tree* $\mathcal{T}_\mathcal{S}$, i.e., finite non-empty graph which is a tree, whose vertices, called *membranes*, are labeled by multisets over the *sets* $\mathbb{O}_\mathcal{S}$ *of objects* of $\mathcal{S}$. More precisely, there is given *labeling function* $\mathcal{M}_\mathcal{S} : V(\mathcal{T}_\mathcal{S}) \to N^{\mathbb{O}_\mathcal{S}}$ of $\mathcal{S}$ defined on the set $V(\mathcal{T}_\mathcal{S})$ of vertices of $\mathcal{T}_\mathcal{S}$ such that the values $\mathcal{M}_\mathcal{S}(v)$ are functions $f : \mathbb{O}_\mathcal{S} \to N$ valued in the set N of natural numbers with 0. For an equivalence of the above treatment membrane systems with the treatment of membrane systems understood as in [23] see Remark 2 in [19].

The comparison of the above models is discussed with regard to the following features of them.

A: basic definitions of systems and computations realized by them are free from concepts involving recursiveness, e.g., recursive families of programs, etc., and

the number of defining axioms and principles is finite and minimal in the sense that leaving one of them does not suffice to prove recursiveness of computational tasks understood as in Introduction;

B: definition of computational complexity measure of consumed space during computation is explicit, natural, and simple;

C: the modeled computations comprise a wide scope of possibilities of parallelism from computations realized by distributed systems, with every processor equipped with an (access) independent memory unit from other processors, to systems with processors sharing an access to common memory unit like in the case of PRAMs in 2;

D: solutions of NP complete problems in a polynomial time with an exponential space expense are provided;

E: immediate extensions to randomized or quantum counterparts are provided, like in [15], [17];

F: immediate extensions to continuous time computations are provided like in [27];

G: explicit treatment of communication (interaction) with environment during computation, understood as in Y. Gurevich's papers ([2], [8]), is provided;

H: the models have an immediate realization by really existing devices (computers) in silicon or biochemical one.

We complete the above listed features A–H by the following comments and remarks containing explanatory, representative, or typical examples.

Ad A. The class of Gandy's machines in 4 is a representative example of a class having the feature A. These machines are defined in an abstract mathematical way in [10] by four principles and [10] contains the result that whatever is computable by the devices satisfying these principles is also computable by Turing machines. The principles are minimal in the sense that no three of them suffice to prove the mentioned result.

Ad B. The class of ordinary membrane machines in 1 is an example of a class having the feature B. Let for an ordinary membrane machine $\Pi = (\Pi^i \,|\, i \in \mathrm{INP})$ a unique evolution process generated by Π^i be presented by the following sequence of length n_i

$$\mathcal{S}_0^i \Rightarrow \mathcal{S}_1^i \Rightarrow \ldots \Rightarrow \mathcal{S}_{n_i}^i,$$

where $\mathcal{S}_0^i, \mathcal{S}_1^i, \ldots, \mathcal{S}_{n_i}^i$ are membrane systems such that $\mathcal{S}_0^i$ is the initial membrane system of the process, $\mathcal{S}_j^i$ evolves into $\mathcal{S}_{j+1}^i$ for all j with $0 \leq j < n_i$, and $\mathcal{S}_{n_i}^i$ is the final membrane system of the process. Then one defines the claimed in B space complexity measure $\mathrm{SPACE}(i)$ by

$$\mathrm{SPACE}(i) = \max_{0 \leq j \leq n_i} \sum_{v \in V(\mathcal{T}_{\mathcal{S}_j^i})} \left(1 + \sum_{a \in \mathbb{O}_{\mathcal{S}_j^i}} \mathcal{M}_{\mathcal{S}_j^i}(v)(a)\right),$$

where $V(\mathcal{T}_{\mathcal{S}_j^i})$, $\mathbb{O}_{\mathcal{S}_j^i}$, $\mathcal{M}_{\mathcal{S}_j^i}$ are the set of vertices of the underlying tree $\mathcal{T}_{\mathcal{S}_j^i}$, the set of objects, and labeling function of $\mathcal{S}_j^i$, respectively, see Remark 2.

Ad C. Some ordinary membrane machines together with evolution processes generated by them can be treated as distributed systems such that every their processor has an independent memory unit, see representative example in Remark 1. In this example a membrane v of an evolving membrane system $\mathcal{S}$ can be treated as a processor equipped with a memory unit containing $\mathcal{M}_{\mathcal{S}}(v)$, see [16] and [18] for more details.

Ad D. The ordinary membrane machine shown in Remark 1 provides a solution of NP complete problem SAT in a polynomial time with an exponential space expense. The construction of this machine was inspired by the pioneering paper [22]. For other related constructions see e.g. [24].

Ad E. Randomized P systems in [17] and quantum P systems in [15] give rise to the extensions of ordinary membrane machines as claimed in E. We point out that one may claim the extensions to randomized counterparts which could provide probabilistic approach to P/NP conjecture proposed in [1].

Ad F. The class of models in 3 is an example of a class of models having feature F because inductive next state formula for neural nets can be simply converted to a system of ordinary differential equations. We point out here that one may claim the extensions to continuous, analog models of computations which could include continuous approach to P/NP conjecture in [5], [6].

Ad G. The attempts to describe the interconnections of membrane systems with environment through their skins are shown among others in [21].

Ad H. The connection machines in 6 have a physical realization. Namely, the Connection Machine Models CM-1, CM-2, and CM-5 were made by Thinking Machines Corporation in Boston, between 1986 and 1996.

3 Concluding Discussion and Open Problems

The comparison of the ordinary membrane machines and the mathematical models with regard to the features described in Section 2 is summarized in the following tables, where new open problems are indicated by question mark "?".

Table of answers to the question:
does X hold for i?

	A	B	C	D	E	F	G	H	?
1	?	Yes	Yes	Yes	Yes	?	Yes?	?	
2	?	Yes	No?	Yes?	Yes	?	No	?	
3	?	?	Yes?	Yes?	Yes	Yes	?	?	
4	Yes	?	Yes?	?	?	?	?	?	
5	?	?	Yes?	Yes?	?	?	Yes	?	
6	?	?	?	?	?	?	?	Yes	
?									

For $X \in \{\mathrm{A}, \mathrm{B}, \ldots, \mathrm{H}\}$, $1 \leq i \leq 6$, and open problems indicated by '?', including conjectures indicated by 'Yes?', 'No?', meant 'rather Yes' and 'rather No', respectively, where the properties $\mathrm{A}, \mathrm{B}, \ldots, \mathrm{H}$ correspond to the comparative features and the numbers $1, 2, \ldots, 6$ correspond to the models as in the lists given in Section 2, respectively.

Matrix-like table of answers to the question:
does i simulate j in polynomial slow-down?
(*i*—row, *j*—column)

	1	2	3	4	5	6
1	Yes	Yes	?	?	?	Yes
2	No?	Yes	No?	No?	No?	Yes
3	?	Yes	Yes	?	?	Yes
4	?	Yes	Yes	Yes	Yes?	Yes
5	?	Yes	?	?	Yes	Yes
6	No	No	No	No	No	Yes

Open problems and conjectures are indicated by '?', 'Yes?', 'No?' as in the first table.

From the first table and its first row we conclude that despite the treatment of natural computing as less important than e.g. multicore computing, cf. [25], or not worth to mention, cf. [3], the bio-inspired membrane computing, a vital part of natural computing, contains ordinary membrane machines which are computation models of advantageous features from complexity theoretical point of view (see features B, C, D) and open for extensions to new approaches to computing outlined among others in [4].

References

1. Arora, S., and Safra, Sh., *Probabilistic checking of proofs: a new characterization of NP*, in: Proc. 33rd IEEE Symp. on Foundation of Computer Science, 1992, 2–12.
2. Blass, A., and Gurevich, Yu., *Algorithms: a quest for absolute definitions*, Bull. Eur. Assoc. Theor. Comput. Sci. EATCS 81 (2003), 195–225.
3. Buss, S. R., Kechris, A. S., Pillay, A., Shore, S. A., *The prospects for mathematical logic in the twenty-first century*, the independent presentations included in panel discussion at The Annual Meeting of the Association for Symbolic Logic held in Urbana–Champaign, June 2000.
4. Calude, C. S., and Păun, Gh., *Bio-steps beyond Turing*, BioSystems 77 (2004), 175–194.
5. Costa, J. F., and Mycka, J., *An analytic condition for $P \subset NP$*, March 2006.
6. Costa, J. F., and Mycka, J., *The conjecture $P \neq NP$ presented by means of some class of real functions*, 2007.
7. Dahlhaus, E., and Makowsky J. A., *Gandy's principles for mechanisms as a model for parallel computation*, in: The Universal Turing Machine: a Half-Century Survey, ed. R. Herken, second ed., Springer, Wien-New York 1995, pp. 283–288.

8. Dershowitz, N., and Gurevich, Yu., *A natural axiomatization of Church's thesis*, July 2007.
9. Fich, F. E., *The Complexity of Computation on the Parallel Random Access Machine*, Chapter 21.
10. Gandy, R., *Church's thesis and principles for mechanisms*, in: The Kleene Symposium, eds. J. Barwise et al., North-Holland, Amsterdam 1980, pp. 123–148.
11. Grover, L., and Rudolph, T., *How significant are the known collision and element distinctness quantum algorithms?*, `arXiv: quant-ph/0309123v1, 16 Sep 2003`.
12. Hillis, W. D., *The Connection Machines*, Cambridge, Mass. 1985.
13. Hirvensalo, M., *Quantum Computing*, second edition, Springer, Berlin 2004.
14. Karp, R. M., and Ramachandran, V., *Parallel algorithms for shared-memory machines*, in: Handbook of Theoretical Computer Science, Vol. A: Algorithms and Complexity, MIT Press, Cambridge 1990, 869–941.
15. Leporati, A., Pescini, D., and Zandron, C., *Quantum energy-based P systems*, in: Proc. Brainstorming Workshop on Uncertainty in Membrane Computing, Palma de Mallorca, November 2004.
16. Obtułowicz, A., *Deterministic P systems for solving SAT problem*, Romanian Journal of Information Science and Technology 4 (2001), 195–201.
17. Obtułowicz, A., *Probabilistic P systems*, in: Membrane Computing, Lecture Notes in Comput. Sci. 2597, Springer, Berlin 2003, 377–387.
18. Obtułowicz, A., *Note on some recursive family of P systems with active membranes*, P systems Web page, 2003.
19. Obtułowicz, A., *Gandy's principles for mechanisms and membrane computing*, in: Proc. Cellular Computing (Complexity Aspects), ESF PESC Exploratory Workshop, January 31–February 2, 2005, ed. M. A. Gutiérrez-Naranjo et al., Sevilla 2005, pp. 267–276.
20. Papadimitriou, Ch. H., *Computational Complexity*, Addison–Wesley, Reading 1994.
21. Păun, A., and Păun, Gh., *The power of communication*: *P systems with symport/antiport*, New Generation Computing 20 (2002), 295–306.
22. Păun, Gh., *P systems with active membranes: Attacking NP complete problems*, Journal of Automata, Languages and Combinatorics 6 (2000), 75–90.
23. Păun, Gh., *Membrane Computing. An Introduction*, Springer-Verlag, Berlin 2002.
24. Perez Jimenez, M. J., Romero Jimenez, A., and Caparrini, F. S., *Decision P systems and $P \neq NP$ conjecture*, in: Membrane Computing, Lecture Notes in Comput. Sci. 2597, Springer, Berlin 2003, 388–399.
25. Scott, D. S., *Looking to the Future*, talk given at CiE (Computability in Europe) Conference, Siena, July 2007, `http://www.mat.unisi.it/~sorbi/sito/CiETalks/ScottCiE07.pdf`.
26. Sieg, W., *Computability Theory*, Seminar Lectures, University of Bologna, November 2004, `http://www.phil.cmu.edu/summerschool/2006/Sieg/computability_theory.pdf`
27. Siegelmann, H. T., and Sontag, E. D., *On the computational power of neural nets*, J. Comput. System Sci. 50 (1995), 132–150.